IN THE CLAIMS

Claims 1-4, 6-15, and 17-37 are amended herein. Claim 5 is canceled. All pending claims are produced below.

1. (Currently Amended) A computer implemented method for in-place preservation of file system objects during a clone operation, the method comprising the steps of:

a cloning manager determining boundaries on a target storage medium, the boundaries to completely contain ~~of~~ a resultant file system to be created by the clone operation;

the cloning manager identifying at least one protected area within the boundaries ~~reserved for the file system~~, the protected area to be ~~created~~ overwritten by the clone operation with a portion of the resultant file system;

the cloning manager identifying at least one in-place file system object at least partially within the boundaries, the in-place file system object not to be overwritten ~~to be preserved~~ during the clone operation;

the cloning manager storing, in a location that will not be affected by the clone operation, metadata concerning each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~;

the cloning manager ensuring that each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ is not located in a protected area; and

the cloning manager performing the clone operation, the clone operation comprising creating the resultant file system ~~during the clone operation~~ only in locations within the boundaries in which no in-place file system object ~~to be preserved~~ is located.

2. (Currently Amended) The method of claim 1 wherein the cloning manager determining the boundaries ~~of a file system to be created by the clone operation~~ comprises:

the cloning manager analyzing data concerning the clone operation to determine at least one attribute concerning the resultant file system to be created from a group of attributes consisting of:

a file system type of the resultant file system to be created;

a location of volume boundaries of the resultant file system to be created;

storage geometry concerning the resultant file system to be created; and a number of total sectors to be used by the resultant file system to be created.

3. (Currently Amended) The method of claim 1 wherein the cloning manager identifying at least one protected area within the boundaries ~~reserved for the file system to be created by the clone operation~~ comprises the cloning manager performing at least one step from a group of steps consisting of:

identifying at least one protected area required by the resultant file system to be created by the clone operation; and

identifying at least one protected area not required by but optimally reserved for the resultant file system to be created by the clone operation.

4. (Currently Amended) The method of claim 1 wherein the cloning manager identifying at least one in-place file system object ~~with the boundaries to be preserved during the clone operation~~ comprises:

the cloning manager compiling a list of in-place file system objects not to be overwritten ~~to be preserved~~ during the clone operation; and

the cloning manager eliminating any in-place file system objects which will not be affected by the clone operation from the list.

5. (Canceled)

6. (Currently Amended) The method of claim 4 wherein the cloning manager eliminating any in-place file system objects which will not be affected by the clone operation from the list further comprises:

the cloning manager identifying at least one in-place file system object ~~to be preserved~~ which is located outside of the boundaries ~~of the file system to be created by the clone operation~~; and

the cloning manager eliminating each identified in-place file system object which is located outside of the boundaries from the list.

7. (Currently Amended) The method of claim 1 wherein the cloning manager storing metadata concerning each in-place file system object ~~to be preserved during the clone operation~~ further comprises:

the cloning manager storing, for each in-place system object ~~to be preserved during the clone operation~~, at least one metadatum concerning the in-place file system object from a group of metadata consisting of:

a path of the in-place file system object;

at least one attribute concerning the in-place file system object; and

a logical location of the in-place file system object;

a physical storage location of content of the in-place file system object.

8. (Currently Amended) The method of claim 1 wherein the cloning manager storing metadata concerning each in-place file system object ~~to be preserved during the clone operation~~ further comprises:

the cloning manager storing the metadata in a location that will not be affected by the clone operation in a format from a group of formats consisting of:

at least two files, each file containing the metadata so as to support fault tolerance;

at least one record in a database supporting fault tolerance;

a single file; and

structured data in random access memory.

9. (Currently Amended) The method of claim 1 wherein the cloning manager ensuring that each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ is not located in a protected area comprises:

the cloning manager comparing a location of each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ to locations of identified protected areas ~~reserved for the file system to be created by the clone operation~~; and

responsive to the cloning manager determining that a location of an in-place file system object conflicts with a location of a protected area, the cloning manager performing a step from a group of steps consisting of:

moving the conflicting in-place file system object to an available non-conflicting location, and updating metadata concerning the in-place file system object accordingly; and

classifying the result of the determination as an error condition.

10. (Currently Amended) The method of claim 1 wherein the ~~cloning manager creating the file system during the~~ clone operation ~~only in locations within the boundaries in which no in-place file system object to be preserved is located~~ further comprises:

before allocating at least one sector for the creation of the resultant file system, ~~the cloning manager~~ checking the stored metadata concerning the in-place file system objects to determine if at least one in-place file system object ~~to be preserved~~ is located at that location; and

responsive to determining that at least one in-place file system object ~~to be preserved~~ is located at that location, allocating the at least one sector to the resultant file system at an available non-conflicting location.

11. (Currently Amended) The method of claim 1 wherein:

the cloning manager identifying at least one in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ further comprises the cloning manager identifying at least one in-place file system object to be ~~both preserved during the clone operation and~~ incorporated into the resultant file system created by the clone operation; and

wherein the cloning manager storing, in a location that will not be affected by the clone operation, metadata concerning each in-place file system object further comprises the cloning manager storing metadata concerning each identified in-place file system object to be ~~both preserved during the clone operation and~~ incorporated into the resultant file system created by the clone operation, the metadata comprising at least one metadatum from a group of metadata consisting of:

an indication that the in-place file system object is to be incorporated in the resultant file system to be created by the clone operation;

a recovery path of the in-place file system object within the resultant file system to be created by the clone operation; and

a recovery partition of the in-place file system object within the resultant file system to be created by the clone operation.

12. (Currently Amended) The method of claim 11 further comprising:

the cloning manager determining that at least one identified in-place file system object to be incorporated into the resultant file system to be created by the clone operation is not compatible with the resultant file system to be created by the clone operation; and

responsive to the determination, the cloning manager performing a step from a group of steps consisting of:

modifying at least one identified in-place file system object to be compatible with the resultant file system to be created by the clone operation; and

classifying the identification as an error condition.

13. (Currently Amended) The method of claim 11 further comprising:

for each identified in-place file system object to be incorporated into the resultant file system, the cloning manager determining whether its content is located within a location that is to be a data area of the resultant file system, and whether its location is ~~properly~~ aligned according to storage geometry of the resultant file system; and

responsive to determining that the location of at least one in-place file system object to be incorporated into the resultant file system is not compatible with the resultant file system, the cloning manager performing a step from a group of steps consisting of:

moving the in-place file system object such that its new location is compatible with the resultant file system and updating the associated metadata accordingly; and

classifying the result of the determination as an error condition.

14. (Currently Amended) The method of claim 11 further comprising the cloning manager performing the following additional steps after the clone operation:

using ~~appropriate~~ stored metadata to create a directory entry in the ~~created~~ resultant file system for each identified in-place file system object to be incorporated into the ~~created~~ resultant file system; and

updating metadata concerning the ~~created~~ resultant file system to map the content location of each identified in-place file system object into the ~~created~~ resultant file system.

15. (Currently Amended) The method of claim 1 further comprising:

the cloning manager determining whether target storage medium is of sufficient size to store each identified in-place file system object ~~to be preserved during the clone operation~~ and the resultant file system to be created by the clone operation;

responsive to the result of the determining step, the cloning manager performing a step from a group of steps consisting of:

responsive to determining that the target storage medium is of sufficient size, proceeding with the clone operation; and

responsive to determining that the target storage medium is not of sufficient size, classifying the result of the determination as an error condition.

16. (Original) The method of claim 1 further comprising:

the cloning manager creating at least two file systems during the clone operation.

17. (Currently Amended) The method of claim 1 further comprising:

the cloning manager creating at least one file system during the clone operation on at least two target storage media.

18. (Currently Amended) A computer readable medium containing a computer program product for in-place preservation of file system objects during a clone operation, the computer program product comprising:

program code for determining boundaries on a target storage medium, the boundaries to completely contain ~~of~~ a resultant file system to be created by the clone operation;

program code for identifying at least one protected area within the boundaries ~~reserved for the file system~~, the protected area to be ~~created~~ overwritten by the clone operation with a portion of the resultant file system;

program code for identifying at least one in-place file system object at least partially within the boundaries, the in-place file system object not to be overwritten ~~to be preserved~~ during the clone operation;

program code for storing, in a location that will not be affected by the clone operation, metadata concerning each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~;

program code for ensuring that each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ is not located in a protected area; and

program code for performing the clone operation, the clone operation comprising creating the resultant file system ~~during the clone operation~~ only in locations within the boundaries in which no in-place file system object ~~to be preserved~~ is located.

19. (Currently Amended) The computer program product of 18 wherein the program code for determining the boundaries ~~of a file system to be created by the clone operation~~ comprises:

program code for analyzing data concerning the clone operation to determine at least one attribute concerning the resultant file system to be created from a group of attributes consisting of:

a file system type of the resultant file system to be created;

a location of volume boundaries of the resultant file system to be created;

storage geometry concerning the resultant file system to be created; and

a number of total sectors to be used by the resultant file system to be created.

20. (Currently Amended) The computer program product of claim 18 wherein the program code for identifying at least one protected area within the boundaries ~~reserved for the file system to be created by the clone operation~~ further comprises at least one program code from a group of program codes consisting of:

program code for identifying at least one protected area required by the resultant file system to be created by the clone operation; and

program code for identifying at least one protected area not required by but optimally reserved for the resultant file system to be created by the clone operation.

21. (Currently Amended) The computer program product of claim 18 wherein the program code for storing metadata concerning each in-place file system object ~~to be preserved during the clone operation~~ further comprises:

program code for storing, for each in-place system object ~~to be preserved during the clone operation~~, at least one metadatum concerning the in-place file system object from a group of metadata consisting of:

a path of the in-place file system object;

at least one attribute concerning the in-place file system object; and

a logical location of the in-place file system object;

a physical storage location of content of the in-place file system object.

22. (Currently Amended) The computer program product of claim 18 wherein the program code for storing metadata concerning each in-place file system object ~~to be preserved during the clone operation~~ further comprises:

program code for storing the metadata in a location that will not be affected by the clone operation in a format from a group of formats consisting of:

at least two files, each file containing the metadata so as to support fault tolerance;

at least one record in a database supporting fault tolerance;

a single file; and

structured data in random access memory.

23. (Currently Amended) The computer program product of claim 18 wherein the program code for ensuring that each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ is not located in a protected area comprises:

program code for comparing a location of each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ to locations of identified protected areas ~~reserved for the file system to be created by the clone operation~~; and

at least one program code for from a group of program codes consisting of:

program code for, responsive to determining that a location of a in-place file system object conflicts with a location of a protected area, moving the conflicting in-place file system object to an available non-conflicting location, and updating metadata concerning the in-place file system object accordingly; and

program code for, responsive to determining that a location of a in-place file system object conflicts with a location of a protected area, classifying the result of the determination as an error condition.

24. (Currently Amended) The computer program product of claim 18 wherein the program code for creating the file system during the clone operation only in locations within the boundaries in which no in-place file system object ~~to be preserved~~ is located further comprises:

program code for, before allocating at least one sector for the creation of the resultant file system, checking the stored metadata concerning the in-place file system objects to determine if at least one in-place file system object ~~to be preserved~~ is located at that location; and program code for, responsive to determining that at least one in-place file system object ~~to be preserved~~ is located at that location, allocating the at least one sector to the resultant file system at an available non-conflicting location.

25. (Currently Amended) The computer program product of claim 18 wherein:

the program code for identifying at least one in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ further comprises program code for identifying at least one in-place file system object to be ~~both preserved during the clone operation and~~ incorporated into the resultant file system created by the clone operation; and

wherein the program code for storing, in a location that will not be affected by the clone operation, metadata concerning each in-place file system object further comprises program code for storing metadata concerning each identified in-place file system object to be ~~both preserved during the clone operation and~~ incorporated into the resultant file system created by the clone operation, the metadata comprising at least one metadatum from a group of metadata consisting of:

an indication that the in-place file system object is to be incorporated in the resultant file system to be created by the clone operation;

a recovery path of the in-place file system object within the resultant file system to be created by the clone operation; and

a recovery partition of the in-place file system object within the resultant file system to be created by the clone operation.

26. (Currently Amended) The computer program product of claim 25 further comprising:

program code for determining, for each identified in-place file system object to be incorporated into the resultant file system, whether its content is located within a location that is to be a data area of the resultant file system, and whether its location is ~~properly~~ aligned according to storage geometry of the resultant file system; and

at least one program code from a group of program codes consisting of:

program code for, responsive to determining that the location of at least one in-place file system object to be incorporated into the resultant file system is not compatible with the resultant file system, moving the in-place file system object such that its new location is compatible with the resultant file system and updating the associated metadata accordingly; and

program code for, responsive to determining that the location of at least one in-place file system object to be incorporated into the file system is not compatible with the file system, classifying the result of the determination as an error condition.

27. (Currently Amended) The computer program product of claim 25 further comprising:

program code for using ~~appropriate~~ stored metadata to create a directory entry in the ~~created~~ resultant file system for each identified in-place file system object to be incorporated into the ~~created~~ resultant file system; and

program code for updating metadata concerning the ~~created~~ resultant file system to map the content location of each identified in-place file system object into the ~~created~~ resultant file system.

28. (Currently Amended) A computer system for in-place preservation of file system objects during a clone operation, the computer system having a computer-readable memory containing a computer program product comprising:

a software portion configured to determine boundaries on a target storage medium, the boundaries to completely contain ~~of~~ a resultant file system to be created by the clone operation;

a software portion configured to identify at least one protected area within the boundaries ~~reserved for the file system~~, the protected area to be ~~created~~ overwritten by the clone operation with a portion of the resultant file system;

a software portion configured to identify at least one in-place file system object at least partially within the boundaries, the in-place file system object not to be overwritten ~~to be preserved~~ during the clone operation;

a software portion configured to store, in a location that will not be affected by the clone operation, metadata concerning each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~;

a software portion configured to ensure that each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ is not located in a protected area; and a software portion configured to perform the clone operation, the clone operation comprising ~~create~~ creating the resultant file system ~~during the clone operation~~ only in locations within the boundaries in which no in-place file system object ~~to be preserved~~ is located.

29. (Currently Amended) The computer system of 28 wherein the software portion configured to determine the boundaries ~~of a file system to be created by the clone operation~~ comprises:

a software portion configured to analyze data concerning the clone operation to determine at least one attribute concerning the resultant file system to be created from a group of attributes consisting of:

a file system type of the resultant file system to be created;
a location of volume boundaries of the resultant file system to be created;
storage geometry concerning the resultant file system to be created; and
a number of total sectors to be used by the resultant file system to be created.

30. (Currently Amended) The computer system of claim 28 wherein the software portion configured to identify at least one protected area within the boundaries ~~reserved for the file system to be created by the clone operation~~ further comprises at least one software portion from group of software portions consisting of:

a software portion configured to identify at least one protected area required by the resultant file system to be created by the clone operation; and

a software portion configured to identify at least one protected area not required by but optimally reserved for resultant the file system to be created by the clone operation.

31. (Currently Amended) The computer system of claim 28 wherein the software portion configured to store metadata concerning each in-place file system object ~~to be preserved during the clone operation~~ further comprises:

a software portion configured to store, for each in-place system object ~~to be preserved during the clone operation~~, at least one metadatum concerning the in-place file system object from a group of metadata consisting of:

a path of the in-place file system object;
at least one attribute concerning the in-place file system object; and
a logical location of the in-place file system object;

a physical storage location of content of the in-place file system object.

32. (Currently Amended) The computer system of claim 28 wherein the software portion configured to store metadata concerning each in-place file system object ~~to be preserved during the clone operation~~ further comprises:

a software portion configured to store the metadata in a location that will not be affected by the clone operation in a format from a group of formats consisting of:

at least two files, each file containing the metadata so as to support fault tolerance;

at least one record in a database supporting fault tolerance;

a single file; and

structured data in random access memory.

33. (Currently Amended) The computer system of claim 28 wherein the software portion configured to ensure that each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ is not located in a protected area comprises:

a software portion configured to compare a location of each in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ to locations of identified protected areas ~~reserved for the file system to be created by the clone operation~~; and

at least one software portion from a group of software portions consisting of:

a software portion configured to move, responsive to determining that a location of an in-place file system object conflicts with a location of a protected area, the conflicting in-place file system object to an available non-conflicting location, and to update metadata concerning the in-place file system object accordingly; and

a software portion configured to classify, responsive to determining that a location of a in-place file system object conflicts with a location of a protected area, the result of the determination as an error condition.

34. (Currently Amended) The computer system of claim 28 wherein the software portion configured to create the file system during the clone operation only in locations within the boundaries in which no in-place file system object ~~to be preserved~~ is located further comprises:

a software portion configured to check, before allocating at least one sector for the creation of the resultant file system, the stored metadata concerning the in-place file system objects to determine if at least one in-place file system object ~~to be preserved~~ is located at that location; and

a software portion configured to allocate, responsive to determining that at least one in-place file system object to be preserved is located at that location, the at least one sector to the resultant file system at an available non-conflicting location.

35. (Currently Amended) The computer system of claim 28 wherein:

the software portion configured to identify at least one in-place file system object at least partially within the boundaries ~~to be preserved during the clone operation~~ further comprises a software portion configured to identify at least one in-place file system object to be ~~both preserved during the clone operation and~~ incorporated into the resultant file system created by the clone operation; and

wherein the software portion configured to store, in a location that will not be affected by the clone operation, metadata concerning each in-place file system object further comprises a software portion configured to store metadata concerning each identified in-place file system object to be ~~both preserved during the clone operation and~~ incorporated into the resultant file system created by the clone operation, the metadata comprising at least one metadatum from a group of metadata consisting of:

an indication that the in-place file system object is to be incorporated in the resultant file system to be created by the clone operation;

a recovery path of the in-place file system object within the resultant file system to be created by the clone operation; and

a recovery partition of the in-place file system object within the resultant file system to be created by the clone operation.

36. (Currently Amended) The computer system of claim 35 further comprising:

a software portion configured to determine, for each identified in-place file system object to be incorporated into the resultant file system, whether its content is located within a location that is to be a data area of the resultant file system, and whether its location is ~~properly~~ aligned according to storage geometry of the resultant file system; and

at least one software portion from a group of software portions consisting of:

a software portion configured to move, responsive to determining that the location of at least one in-place file system object to be incorporated into the resultant file system is not compatible with the resultant file system, the in-place file system object such that its new location is compatible with the resultant file system and updating the associated metadata accordingly; and

a software portion configured to classify, responsive to determining that the location of at least one in-place file system object to be incorporated into the resultant file system is not compatible with the resultant file system, the result of the determination as an error condition.

37. (Currently Amended) The computer system of claim 35 further comprising:

a software portion configured to use ~~appropriate~~ stored metadata to create a directory entry in the ~~created~~ resultant file system for each identified in-place file system object to be incorporated into the ~~created~~ resultant file system; and

a software portion configured to update metadata concerning the ~~created~~ resultant file system to map the content location of each identified in-place file system object into the ~~created~~ resultant file system.